TRANSCODING SCHEME FOR ASSISTANCE IN IMAGE TRANSFORMATIONS

Venkat V. Easwar

Vivek K. Thakur

TECHNICAL FIELD OF THE INVENTION

The technical field of this invention is printers and more particularly conversion of print data in a page description language into print drive signals, a process
5 called raster image processing.

BACKGROUND OF THE INVENTION

There is a problem of transforming an image. such as by scale, rotate, translate, etc., when the original image is in
10 a compressed format. If the printer has limited memory., the

image has to be transformed in-situ, because there is no space
available to decompress the image fully for transformation.

Image transformation is a common operation in postscript
based printers. The transformation problem arises even in the
5   simple case where an image in portrait mode is to be printed
in the landscape mode. This corresponds to a 90° rotation
transformation and is a common operation in printing. The
problem is exacerbated by the fact that current compression
methods, like Joint Photographers Expert Group (JPEG), are not
10 designed for random access. Image transformation schemes
often must address the source image in a random fashion. In
such compressed formats this requires multiple decoding passes
through the compressed source image.

15 SUMMARY OF THE INVENTION

This invention proposes to transcode the compressed
image, that may be in the JPEG format for example, to an
intermediate format that allows pseudo-random access. Such a
pseudo-random access would that allow efficient image
20 transformation. By using this format, in most cases a pixel
is decoded only once in the entire image transformation
process. This is certainly true for the most common
transformation operations such as rotation by 90, 180 and 270
degrees. This transcoding would enable image transformations
25 in printers whose memory is insufficient to store the entire
decompressed image.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects of this invention are illustrated
30 in the drawings, in which:

- 2 -

Figure 1 illustrates the source and destination of an example image transformation;

Figure 2 illustrates the typical compressed image coding path and the coding path of this invention;

5　　　　Figure 3 is a flow chart illustrating the process of transcoding according to this invention;

Figure 4 is a flow chart illustrating image transformation employing the transcoded image of this invention;

10　　　Figure 5 illustrates a further example of image transformation using the techniques of this invention; and

Figure 18 shows a block diagram of the TMS320C82 DSP

15　　DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Image transformation is a common task in raster image processing, that is changing print data from a page description language such as postscript to printer control data. An example of image transformation is rotating a source

20　　image that is in portrait mode to a landscape mode. In current art, image transformation methods assume ·that the source image is in an uncompressed form. Any compressed image must be fully decompressed before the transformation. This approach will not work for large images when there is limited

25　　memory. So there is a need for an image transformation scheme that works directly on compressed images.

Figure 1 illustrates an image in source coordinates 110 and destination coordinates 120 of an example image rotation.

The problem with popular compression schemes like JPEG is

30　　that they do not permit random-access. To access a particular

pixel, the entire bitstream before that pixel has to be decoded. This creates a problem for image transformation as illustrated in Figure 1. Scanline 121 from initial pixel A_{XY} to pixel B_{XY} in destination coordinates 120 does not map to a

5   scanline in source coordinates 110. Note scanline 121 corresponds to an oblique line 111 in source coordinates 110.

Because the source image is sequentially coded in JPEG, it is necessary to decode more pixels in source coordinates 110 than are used in the actual mapping to destination coordinates 120.

10   In the prior art the entire source image is decoded in preparation for transformation or a transformation technique is used that maps a source pixel to one or more destination pixels rather than mapping a destination pixel from source.

If there is insufficient memory to store the entire

15   uncompressed image the first technique will not work. If a banding approach is used to conserve memory space, the latter technique requires multiple decoding passes through the source. This is computationally expensive.

This invention proposes to define a new compression

20   scheme that can provide pseudo-random access to individual pixels. Previously this required that a source image already coded in JPEG needed to be fully decompressed and compressed in the new format. This is a computationally time-consuming task. This invention proposes a more efficient scheme that

25   transcodes the JPEG source image into a pseudo-random access JPEG-like compressed formal. The advantage of this approach is that the source image need not be fully decoded and coded again, but is rather transcoded, thereby saving computational time. Further, since the new format closely resembles JPEG

30   much of the same hardware, such as discrete cosine

- 4 -

transformers, variable length decoders and the like, can be used to accelerate the compression/decompression process.

The main advantage of this invention is that there is no need to provide memory to store a fully decompressed source image. Since the image may be the entire page, this could require a large amount of memory to store. In addition, the pseudo-random access capability eliminates or substantially reduces multiple decoding passes. This reduces computation time. The transcoding method has the advantage that the source image is not fully decoded and recoded, so the penalty of recoding JPEG compressed images is minimized.

In this invention, an image is divided into cells that are coded independently and the starting positions of the cells is stored in a pseudo-random access table. The image transformation algorithm decodes a cell when needed and discards it when a pixel in another cell is accessed.

This technique has many advantages. Rotation by 90° is a very common operation on printers which print a page in landscape mode. The invention requires effectively only one decompression cycle for compressed source images. Without the invention such image transformation may require: significant memory for storing the uncompressed image for rotation; significant processing time for JPEG compressed images, which permit only sequential access, including more than one decompression cycle. The invention facilitates using a banding approach if the input image is in a compressed format, by reducing the compression/decompression complexity. This invention has the advantage that the JPEG bitstream need not be fully decoded and coded into the new format. Instead the bitstream is only partially decoded.

Translation, rotation and scaling are typical operations in image transformation. Assuming a translation of (t_u, t_v) followed by a rotation by the angle θ, followed by a scaling of s_u and s_v, the corresponding coordinate transformation can

5   be represented as a single matrix M that transforms source pixel coordinates (u, v) to destination coordinates (x, y):

destination coordinates[x,y,1] = source coordinates[u,v,1]HM

10 where: M is

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_u & t_v & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} s_u & 0 & 0 \\ 0 & s_v & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} s_u \bullet \cos\theta & s_v \bullet \sin\theta & 0 \\ -s_u \bullet \sin\theta & s_v \bullet \cos\theta & 0 \\ s_u \bullet \left(t_u \bullet \cos\theta - t_v \bullet \sin\theta \right) & s_v \bullet \left(t_u \bullet \sin\theta + t_v \bullet \cos\theta \right) & 1 \end{bmatrix}$$

The inverse mapping is just the matrix inverse M^{-1}.

15

An example of image transformation is illustrated in Figure 1. Image abcd is transformed to a'b'c'd'. Scanline A_{xy}-B_{xy} in the destination coordinates 120 does not map to a scanline in the source coordinate 110, but rather to the

20 slanted line A_{uv}-B_{uv} that intersects several scanlines. If the initial image is compressed and there is no memory to expand the image, then this remapping requires repeated

compression/decompression cycles. This invention minimizes these cycles by transcoding the JPEG format to an intermediate format that is JPEG compatible and that also includes a table for pseudo-random access of cells.

5   Figure 2 illustrates an example of the transcoding of this invention. Figure 2 shows an original image 210 divided into cells that are integer multiples of a basic 8x8 blocks used in JPEG. As shown in Figure 2, the basic block are traversed in a raster scan fashion as indicated by path 211.

10   The remaining lines are traversed in the same fashion. In the alternate compressed format 220 the image is divided into plural blocks A through L. Each of these superblocks is traversed in raster scan fashion as shown in path 221. Each additional line within each superblock A to L is traversed in

15 the same fashion. The alternative compressed format 220 includes a header that gives the cell width and height. This header also includes a pseudo-random access table that stores the address locations of the top-left corners of each cell as 64 plus 6-bit number assuming a 64-bit address bus. The extra

20 6-bits are needed because the cell start point need not fall on byte boundaries. An example pseudo-random access table of the header is listed below in Table 1.

Cell	Byte Address (Hex)	Bit position
A	00000000	6
B	00000200	2
C	00000350	3
D	00000500	0
E	00000700	1
...	...	...
L	0000A000	4

Table 1

5   The header may also include a Huffman table. An example is
listed below in Table 2.

1	10000000
2	10000202
3	10000450
4	10000520
5	10000780
...	...
n	1000F000

Table 2

5   Given a particular destination pixel, its corresponding
address in the source image can be calculated using the
coordinate transformation M^{-1}. Its corresponding cell can also
be calculated using a similar transformation. If this cell is
not currently resident in memory in a decoded format, it is
10   decoded and the current cell may be discarded. Any suitable
interpolation scheme, such as nearest neighbor, bilinear,
etc., can be used to determine to the gray level at the device
pixel.

The JPEG format is a popular for encoding images. The
15   JPEG format is used in page description languages such as
postscript. In the baseline JPEG technique, the DC frequency
coefficients of the 8x8 blocks are differential pulse code
modulation (DPCM) coded. The delta difference from the
previous block's DC coefficient is coded. The JPEG format

allows insertion of restart intervals that break this
sequential coding. Such restart intervals can appear at
arbitrary points in the bitstream. So to determine a gray
level at a particular location, the entire bitstream for that
5 image before that location may need to be decoded.

This invention segments the JPEG coded image into
independently coded cells as illustrated in Figure 2. The
invention transcodes these cells into a format that is
randomly accessible at the cell level. This is called
10 pseudo-random access. A flow chart of the algorithm is
illustrated in Figure 3. Process 300 begins at start block
301. Next process 300 decodes the DPCM DC coefficients of the
JPEG bitstream (block 302). These coefficients are recoded
with DPCM at block 303, but DPCM coding is not allowed across
15 the cell boundaries A to L illustrated in Figure 2. Process
300 next extracts all the Huffman tables from the bitstream
(block 304). These Huffman tables are saved elsewhere with an
associated numbering. For each cell, the corresponding
Huffman table number is stored in its header. If new Huffman
20 tables are used within a cell, that information is signaled
with the standard JPEG define Huffman table (DHT) marker
followed by the table number. Process 300 then identifies the
blocks by a block count (block 305). This is maintained by
identification of ends of blocks. This will require ability
25 to find boundaries of the variable length codes, but does not
require using the Huffman tables. Process 300 next recodes
the bitstream into cells (block 306). This includes
separately storing the corresponding starting address for each
cell (block 307). Process 300 is then complete (end block
30 308).

- 10 -

A particular advantage of this transcoding technique is
what is not required. A full decompression of the compressed
image typically would require an Inverse Discrete Cosine
Transform (IDCT). Recoding from a fully decoded image would
5   require a corresponding Discrete Cosine Transform (DCT). The
DCT and IDCT processes are multiply intensive and would
require a large amount of processor time and capacity. The
compressed image need not be fully decoded but is only
partially decoded and recoded into the new format. Because
10   the transcoding of this invention does not fully decode the
compressed image, much otherwise necessary computation is
avoided.

Figure 4 illustrates an example of an image
transformation using the transcoding of this invention.
15   Consider the scan line 111 (A_{uv}-B_{uv}) shown in Figure 1. This
traverses several cells in the input image. The cells that
are traversed can be determined because the transformation
matrix is known and so are the scan line coordinates. When
the current pixel crosses a cell boundary, the next cell in
20   the list is decompressed. Depending on memory constraints,
the previous active cell discarded or temporarily retained.

The process 400 begins at start block 401. Process 400
first identified the next source pixel in the image
transformation (block 402). Process 400 tests to determined
25   if this next source pixel is in a new cell (block 403). If
not, then process 400 processes the image transformation
(block 404). Process 400 then tests to determine if this is
the last source pixel (block 405). If so, then process 400
ends at end block 405). If not, process 400 returns to block
30   402 for the next source pixel.

- 11 -

If the next source pixel was in a new cell (yes at block 403), then process 400 tests to determine if this new cell is in memory (block 407). It is assumed that there is insufficient memory to store the whole decompressed image.

5   However, there may be enough memory allocated to store several cells in decompressed form. If such memory is available, the new cell may already be stored in the memory (yes at block 407). If so, process 400 proceeds with the image transformation (block 404). If the new cell is not in memory

10   (no at block 407), then process 400 tests to determine if the memory is full (block 408). If the memory is not full (no at block 408), meaning that enough memory if free to store another decompressed cell, then process 400 decompressed the new cell (block 410) storing the result in the available

15   memory. Process 400 then proceeds with the image transformation (block 404). If the memory is full (yes at block 408), them process 400 discards an old cell (block 409), decompresses the new cell (block 410) and proceeds with the image transformation (block 404). In the event that only

20   enough memory for one decompressed cell is available, then the new cell is never in memory (no at block 407) and the memory is always full (yes at block 408). Thus in this case, the old cell is discarded (block 409) before decompression of the new cell (block 410).

25   Figure 5 illustrates an alternative technique called the scancell approach. This alternative technique remaps a scancell at a time. Thus the scancell L 511 in the source image 510 is decompressed, scaled, and rotated and mapped to scancell L 521 in the destination image 520 using a bit block

30   transfer (BIT-BLT) operation. This approach is especially

- 12 -

advantageous for the 0, 90, 180 and 270 degree rotation cases as it ensures that no cell is decompressed twice.

In the case of nearest neighbor interpolation method, one source pixel may map to several device pixels. This can occur
5   because the source is usually at a lower resolution than the output device. For example, the source image can come from a 300 dots per inch (dpi) scanner whereas the printer resolution can be 600 dpi. In this case it may be advantageous to access the source pixels and for each source pixel determine all its
10 destination pixels. The random access provided by the transcoding scheme is also an advantage here. For example referring back to Figure 5, when destination block L is desired, the source block L is readily available because of the random-access capability.
15   Suppose the input image is uncompressed and not enough memory is available to store it. In this case the image can be compressed on the fly into the format discussed above. Then the techniques of this invention can be applied to image transformation. If the input image is a higher dpi than the
20 output, the input image can be subsampled on the fly.

Figure 6 illustrates a block diagram of a TMS320C82 digital signal processor (DSP) in an image data processing system capable of practicing this invention. The multiprocessor DSP is a single integrated circuit 180.
25 Integrated circuit 180 a fully programmable parallel processing platform that integrates two advanced DSP cores DSP 181 and DSP 182, a reduced instruction set computer (RISC) master processor (MP) 183, multiple static random access memory (SRAM) blocks 185, 186 and 187, a crossbar switch 184
30 that interconnects all the internal processors and memories,

and a transfer controller (TC) 188 that controls external communications. Transfer controller 188 is coupled to image memory 190 via bus 195. Note that transfer controller 188 controls all data transfer between integrated circuit 180 and

5   image memory 190.